ART Neural Networks for Remote Sensing: Vegetation Classification from Landsat TM and Terrain Data

Gail A. Carpenter, Marin N. Gjaja, Sucharita Gopal, and Curtis E. Woodcock

***Abstract*— A new methodology for automatic mapping from Landsat thematic mapper (TM) and terrain data, based on the fuzzy ARTMAP neural network, is developed. System capabilities are tested on a challenging remote sensing classification problem, using spectral and terrain features for vegetation classification in the Cleveland National Forest. After training at the pixel level, system performance is tested at the stand level, using sites not seen during training. Results are compared to those of maximum likelihood classifiers, as well as back propagation neural networks and K Nearest Neighbor algorithms. ARTMAP dynamics are fast, stable, and scalable, overcoming common limitations of back propagation. Best results are obtained using a hybrid system based on a convex combination of fuzzy ARTMAP and maximum likelihood predictions. A prototype remote sensing example introduces each aspect of data processing and fuzzy ARTMAP classification. The example shows how the network automatically constructs a minimal number of recognition categories to meet accuracy criteria. A voting strategy improves prediction and assigns confidence estimates by training the system several times on different orderings of an input set.**

I. Introduction: Neural Networks and Remote Sensing

MAPPING vegetation from satellite remote sensing data has been an active area of research and development over the past 20 years [1], [2], and neural networks have recently been successfully applied to this problem. Data sources that have been analyzed with neural networks include the Landsat Multispectral Scanner (MSS) [3], Landsat Thematic Mapper (TM) [4]–[6], SPOT (Systeme Pour l'Observation de la Terre) [7], synthetic aperture radar (SAR) [8], [9], Advanced Very High Resolution Radiometer (AVHRR) [10], and multidirectional Advanced Solid-State Array Spectroradiometer (ASAS) [11]. Classification studies that seek to identify landcover classes range from broad life-form categories [12] to narrow floristic classes [13]. In supervised learning studies, input presented to a neural network during training typically consists of spectral data [14], and output consists of ground truth information about a vegetation class, although multispectral image information alone has sometimes proven insufficient for differentiating species-level vegetation classes. Many factors contribute to this problem, including the effects of local topography, background reflectance from soils or understory vegetation, high within-class variance due to the structure and patchiness of vegetation canopies, and the limitations of classification methodologies. To help differentiate vegetation types at the species level, ancillary data have often been used, and it is now common to use topographic variables such as elevation, slope, and aspect in predictive models [15]–[17]. Mapping systems that use spectral and ancillary data typically resemble rule-based expert systems [18]–[20].

Neural networks can improve classification accuracy by 10–30% compared to conventional classification techniques. Back propagation [21], [22], a feedforward multilayer perceptron [23], has been used in a large majority of these studies. Other neural network applications employ the binary diamond network [24], fuzzy ARTMAP [10], and ART [4]. Research on classification methods for remote sensing, including neural networks, also continues [3], [25]–[28]. In general, these studies show that: a) neural network classifiers, which make no *a priori* assumptions about data distributions, are able to learn nonlinear and discontinuous data samples; b) neural networks can readily accommodate ancillary data such as textural information, slope, aspect, and elevation; c) neural networks are typically more accurate than conventional classifiers; and d) neural network architectures are quite flexible and can be adapted to improve performance on particular problems.

The fuzzy ARTMAP neural network is here presented as the basis of a systematic methodology for automatic classification of vegetation at the species level from multispectral and ancillary data. Section II introduces the ART and ARTMAP neural networks and Sections III–V provide self-contained descriptions of fuzzy ART and fuzzy ARTMAP, including a complete implementation algorithm. A prototype remote sensing example (Section VI) illustrates fuzzy ARTMAP dynamics (Section VII). A series of tests then compare fuzzy ARTMAP properties and predictions with those of a maximum likelihood classifier, as well as K Nearest Neighbor and back propagation algorithms (Section VIII). Inputs specify Landsat TM and terrain data from the Cleveland National Forest. During testing, pixel-level predictions are pooled to give a vegetation class prediction of a small region, or site. Test set performance statistics are measured at sites not seen during training. Results show that the fuzzy ARTMAP neural network performs well on a series of difficult remote sensing

Manuscript received October 3, 1995; revised October 9, 1996. This work was supported in part by the National Science Foundation (NSF IRI 94-01659 and NSF SBR 93-00633) and the Office of Naval Research (ONR N00014-95-1-0409 and ONR N00014-95-0657).

G. A. Carpenter and M. N. Gjaja are with the Center for Adaptive Systems and Department of Cognitive and Neural Systems, Boston University, Boston, MA 02215 USA.

S. Gopal and C. E. Woodcock are with the Center for Remote Sensing and Department of Geography, Boston University, Boston, MA 02215 USA.

Publisher Item Identifier S 0196-2892(97)02101-3.

problems. Because fuzzy ARTMAP and maximum likelihood make predictive errors at different locations, a hybrid system can be constructed to give optimal performance. The study defines a general purpose methodology for automatic map construction from remote sensing and ancillary data.

II. ART and ARTMAP Neural Networks

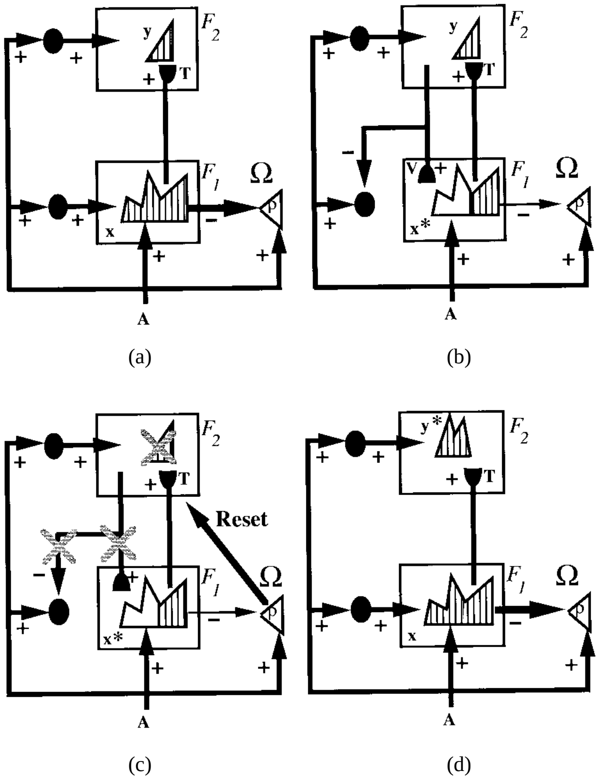
Adaptive resonance theory (ART), introduced in the 1970s as a theory of human cognitive information processing [29], has led to an evolving series of real-time neural network models for unsupervised and supervised category learning and pattern recognition. These models form stable recognition categories in response to arbitrary input sequences with either fast or slow learning regimes. The first ART model, ART 1 [30], was an unsupervised learning system to categorize binary input patterns. ART 1 and subsequent models added new concepts to the theory and have been used for a wide variety of scientific and technological applications [31]. ART 2 [32] and fuzzy ART [33] extend the binary ART 1 domain to categorize both analog and binary input patterns.

A class of supervised network architectures, called ARTMAP systems, self-organize arbitrary mappings from input vectors, representing features such as spectral values and terrain variables, to output vectors, representing predictions such as vegetation classes or mixtures. ARTMAP's internal control mechanisms create stable recognition categories of optimal size by maximizing code compression while minimizing predictive error in an on-line setting. Binary ART 1 computations are the foundation of the first ARTMAP network [34], which therefore learns binary maps. When fuzzy ART replaces ART 1 in an ARTMAP system, the resulting fuzzy ARTMAP architecture [35] rapidly learns stable mappings between analog or binary input and output vectors. This article demonstrates fuzzy ARTMAP performance on a difficult remote sensing problem (Section VIII). A simplified version of this problem (Sections VI and VII) introduces and illustrates fuzzy ARTMAP networks and also summarizes the data processing methods developed for remote sensing applications.

A. ART

The central feature of all ART systems is a pattern matching process that compares the current input with a selected learned category representation, or active hypothesis. This matching process leads either to a resonant state that focuses attention and triggers category learning or to a self-regulating parallel memory search that is guaranteed to lead to a resonant state, unless the network's memory capacity is exceeded. If the search ends with selection of an established category, then the category's learned representation may be refined to incorporate new information from the current input. If the search ends by selecting a previously untrained node, the ART network establishes a new category.

Fig. 1 illustrates the ART search cycle. During ART search, an input vector $\mathbf{A}$ registers itself as a pattern $\mathbf{x}$ of short-term memory (STM) activity across level F_1 [Fig. 1(a)]. Converging and diverging $F_1 \rightarrow F_2$ adaptive filter pathways, each weighted by a long term memory (LTM) trace, or adaptive weight, transform $\mathbf{x}$ into a net input vector $\mathbf{T}$ to level F_2. The internal competitive dynamics of F_2 contrast-enhance vector $\mathbf{T}$, generating a compressed activity vector $\mathbf{y}$ across F_2. In ART 1 and fuzzy ART, strong competition selects the F_2 node that receives the maximal $F_1 \rightarrow F_2$ input component T_J. Only one component (y_J) of $\mathbf{y}$ remains positive after this choice takes place. Activation of such a winner-take-all node selects category J for the input pattern $\mathbf{A}$.



(a) (b) (c) (d)

Fig. 1. ART search for an F_2 code: (a) The input vector $\mathbf{A}$ generates the F_1 activity vector $\mathbf{x}$ as it activates the orienting subsystem Ω. Activity $\mathbf{x}$ both inhibits Ω and generates an $F_1 \rightarrow F_2$ signal. A bottom-up adaptive filter transforms $\mathbf{x}$ into the F_2 input vector $\mathbf{T}$, which activates the STM pattern $\mathbf{y}$ across F_2. (b) A top-down adaptive filter transforms $\mathbf{y}$ into the category representation vector $\mathbf{V}$. Where $\mathbf{V}$ mismatches $\mathbf{A}$, F_1 registers a diminished STM activity pattern $\mathbf{x}^*$. The resulting reduction of total STM reduces the total inhibitory signal from F_1 to Ω. (c) If the ART matching criterion fails, Ω releases a nonspecific signal that resets the STM pattern $\mathbf{y}$ at F_2. (d) Since reset inhibits $\mathbf{y}$, it also eliminates the top-down signal $\mathbf{V}$, so $\mathbf{x}$ can be reinstated at F_1. However, enduring traces of the prior reset allow $\mathbf{x}$ to activate a different STM pattern $\mathbf{y}^*$ at F_2. If the top-down signal due to $\mathbf{y}^*$ also mismatches $\mathbf{A}$ at F_1, then the search for an F_2 code that satisfies the matching criterion continues [30].

Activation of an F_2 node may be interpreted as "making a hypothesis" about an input $\mathbf{A}$. After sending the F_2 activity vector $\mathbf{y}$ through top-down adaptive filter pathways, a filtered vector $\mathbf{V}$ becomes the $F_2 \rightarrow F_1$ input [Fig. 1(b)]. The ART network matches the "expectation" pattern $\mathbf{V}$ of the active category against the current input pattern, or exemplar, $\mathbf{A}$. This matching process typically changes the F_1 activity pattern $\mathbf{x}$, suppressing activation of all features in $\mathbf{A}$ that are not confirmed by $\mathbf{V}$. The resultant pattern $\mathbf{x}^*$ represents the features to which the network "pays attention." If the expectation $\mathbf{V}$ is close enough to the input $\mathbf{A}$, then a state of resonance occurs, with the matched pattern $\mathbf{x}^*$ defining an attentional focus. The resonant state persists long enough for weight adaptation to occur; hence, the term *adaptive resonance* theory. The fact that ART networks encode only attended features $\mathbf{x}^*$ rather than all input features $\mathbf{A}$ is directly responsible for ART code stability. This characteristic differentiates ART from feedforward neural networks, which typically encode the current vector $\mathbf{A}$, rather than a matched pattern, and hence require slow learning to avoid catastrophic forgetting of past memories.

A dimensionless parameter called *vigilance* defines the criterion of an acceptable match. Vigilance specifies what fraction of the bottom-up input $\mathbf{A}$ must remain in the matched F_1 pattern $\mathbf{x}^*$ in order for resonance to occur. In unsupervised ART systems, vigilance is a fixed parameter, but in ARTMAP, vigilance becomes an internally controlled variable. Because vigilance then varies across learning trials, a single ARTMAP system can encode widely differing degrees of generalization, or code compression. Low vigilance allows broad generalization, coarse categories, and abstract representations. High vigilance leads to narrow generalization, fine categories, and specific representations. At the very high vigilance limit, category learning reduces to exemplar learning. Varying vigilance levels allow a single ART system to recognize both abstract categories, such as faces and dogs, and individual faces and dogs.

ART memory search, or hypothesis testing, begins when the top-down expectation $\mathbf{V}$ determines that the bottom-up input $\mathbf{A}$ is too novel, or unexpected, with respect to the chosen category to satisfy the vigilance criterion. Search leads to selection of a better recognition code to represent input $\mathbf{A}$ at level F_2. An *orienting subsystem* Ω controls the search process. The orienting subsystem interacts with the attentional subsystem [Fig. 1(b) and (c)] to enable the network to learn about novel inputs without risking unselective forgetting of its previous knowledge. ART 3 [36] implements parallel distributed search as a medium-term memory (MTM) process, as needed for distributed recognition codes.

ART search prevents associations from forming between $\mathbf{y}$ and $\mathbf{x}^*$ if $\mathbf{x}^*$ is too different from $\mathbf{A}$ to satisfy the vigilance criterion. The search process resets $\mathbf{y}$ before such an association can form. If the vigilance criterion is met, then the active category's representation may be refined in light of new information carried by $\mathbf{A}$. If the search ends upon an uncommitted F_2 node, then $\mathbf{A}$ begins a new category. An ART *choice parameter* α controls how deeply the search proceeds before selecting an uncommitted node. In a parameter range called the *conservative limit*, where α is very small, an input first selects a category whose weight vector is a subset of the input, if such a category exists. Given such a choice, no weight change occurs during learning; hence the name conservative limit, since learned weights are conserved wherever possible. As learning self-stabilizes, all inputs coded by a category access it directly, search is automatically disengaged, and the performance rate reaches 100% on the training set.

Many ART applications use fast learning, whereby adaptive weights fully converge to equilibrium values in response to each input pattern. Fast learning enables a system to adapt quickly to inputs that occur only rarely but that may require immediate accurate performance. Remembering many details of an exciting movie is a typical example of fast learning. Fast learning destabilizes the memories of feedforward, error-based models like back propagation. When the difference between actual output and target output defines "error," present inputs drive out past learning, since fast learning zeroes the error on each input trial. This feature of back propagation restricts its domain to off-line applications with a slow learning rate. In addition, lacking the key feature of competition, a back propagation system tends to average rare events with similar frequent events that have different consequences.

Some applications benefit from a *fast-commit slow-recode* option that combines fast initial learning with a slower rate of forgetting. Fast commitment retains the advantage of fast learning, namely, the ability to respond to important distinctive inputs that occur only rarely. Slow recoding then prevents features in a category's learned representation from being erroneously altered in response to noisy or partial inputs.

Complement coding is a preprocessing step that normalizes input patterns and solves a potential fuzzy ART category proliferation problem [33], [37]. In neurobiological terms, complement coding uses both on-cells and off-cells to represent an input pattern, preserving individual feature amplitudes while normalizing the total on-cell/off-cell activity. Functionally, the on-cell portion of a weight vector encodes features that are consistently present in category exemplars, while the off-cell portion encodes features that are consistently absent. Small weights in both on-cell and off-cell portions of a category representation encode as "uninformative" those features that are sometimes present and sometimes absent. Complement coding allows a geometric interpretation of fuzzy ART recognition categories as box-shaped regions of input space. Tests of a prototype remote sensing example illustrate fuzzy ART geometry with inputs that provide two TM spectral band values at each pixel (Section VII). Thus the inputs are two-dimensional and category boxes are rectangles.

B. ARTMAP

Each ARTMAP system includes a pair of ART modules (ART_a and ART_b) that create stable recognition categories in response to arbitrary sequences of input patterns (Fig. 2). During supervised learning, ART_a receives a stream of patterns $\{\mathbf{a}^{(n)}\}$ and ART_b receives a stream of patterns $\{\mathbf{b}^{(n)}\}$, where $\mathbf{b}^{(n)}$ is the correct prediction given $\mathbf{a}^{(n)}$. An associative learning network and an internal controller link these modules to make the ARTMAP system operate in real time. The controller creates the minimal number of ART_a recognition categories, or "hidden units," needed to meet accuracy criteria. A minimax learning rule enables ARTMAP to learn quickly, efficiently, and accurately as it conjointly minimizes predictive error and maximizes code compression. This scheme automatically links predictive success to category size on a trial-by-trial basis using only local operations. It works by increasing the ART_a vigilance parameter $\rho = \rho_a$ by the minimal amount needed to correct a predictive error at ART_b.

A *baseline vigilance* parameter $\bar{\rho} = \bar{\rho}_a$ calibrates a minimum confidence level at which ART_a will accept a chosen category. Lower values of $\bar{\rho}$ allow larger categories to form, maximizing code compression. Initially, $\rho = \bar{\rho}$.

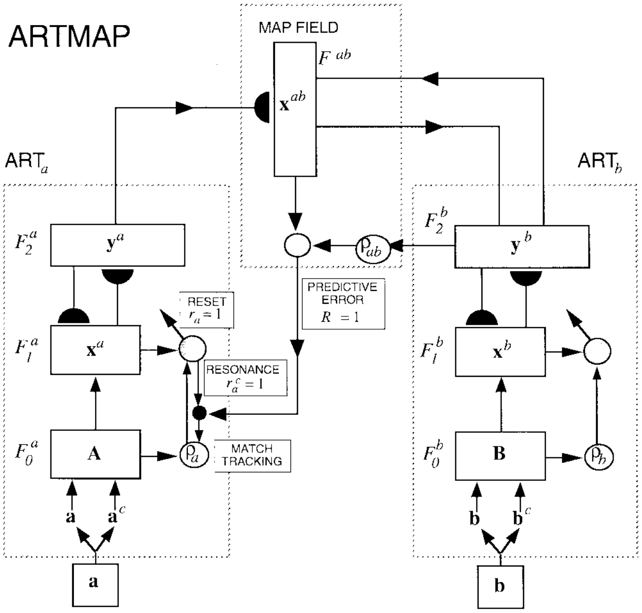


Fig. 2. ARTMAP architecture. The ART_a complement coding preprocessor transforms the M_a-vector $\mathbf{a}$ into the $2M_a$-vector $\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$ at the ART_a field F_0^a. $\mathbf{A}$ is the input vector to the ART_a field F_1^a. Similarly, the input to F_1^b is the $2M_b$-vector $\mathbf{B} = (\mathbf{b}, \mathbf{b}^c)$. When ART_b disconfirms a prediction of ART_a, map field inhibition induces the match tracking process. Match tracking raises the ART_a vigilance ρ_a to just above the F_1^a-to-F_0^a match ratio $|\mathbf{x}^a|/|\mathbf{A}|$. This triggers an ART_a search which leads either to an ART_a category that correctly predicts $\mathbf{b}$ or to a previously uncommitted ART_a category node [34].

During training, a predictive failure at ART_b increases ρ just enough to trigger an ART_a search, through a feedback control mechanism called *match tracking* [34]. Match tracking sacrifices the minimum amount of compression necessary to correct the predictive error. Hypothesis testing selects a new ART category, which focuses attention on a cluster of $\mathbf{a}^{(n)}$ input features that is better able to predict the output $\mathbf{b}^{(n)}$. With fast learning, match tracking allows a single ARTMAP system to learn a different prediction for a rare event than for a cloud of similar frequent events in which it is embedded.

III. Fuzzy ART Dynamics

This section summarizes key features of fuzzy ART dynamics, with a complement coding preprocessor.

A. Field Activity Vectors

A fuzzy ART system includes a field F_0 of nodes that represent a current input vector; a field F_2 that represents the active code, or category; and a field F_1 that receives both bottom-up input from F_0 and top-down input from F_2. Vector $\mathbf{A}$ denotes F_0 activity, with each component A_i in the interval [0, 1]. With complement coding, $\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$. That is, $A_i = a_i$ for $i = 1 \ldots M$; and $A_i = a_{i-M}^c \equiv (1 - a_{i-M})$ for $i = M+1 \ldots 2M$. Vector $\mathbf{x} = (x_1, \ldots, x_{2M})$ denotes F_1 activity and $\mathbf{y} = (y_1, \ldots, y_N)$ denotes F_2 activity. The number of input components (M) and the number of category nodes (N) can be arbitrarily large.

B. Weight Vector

Associated with each F_2 category node $j (j = 1 \ldots N)$ is a vector $\mathbf{w}_j \equiv (w_{1j}, \ldots, w_{2M,j})$ of adaptive weights, or long-term memory (LTM) traces. Initially

$$w_{1j}(0) = \cdots = w_{ij}(0) = \cdots = w_{2M,j}(0) = 1. \qquad (1)$$

Then each category is *uncommitted*. After a category codes its first input, it becomes *committed*. Each component w_{ij} can decrease toward 0 but never increase during learning, so weights always converge. The fuzzy ART weight vector $\mathbf{w}_j$ denotes both the bottom-up and the top-down weight vectors of ART 1.

C. Parameters

A choice parameter $\alpha > 0$, a learning rate parameter $\beta \in [0, 1]$, and a vigilance parameter $\rho \in [0, 1]$ determine fuzzy ART dynamics.

D. Category Choice

For each input $\mathbf{a}$ and F_2 node j, the *choice function* T_j is defined by

$$T_j(\mathbf{A}) = \frac{|\mathbf{A} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|} \qquad (2)$$

where the fuzzy intersection $\wedge$ [38] is defined by

$$(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i) \qquad (3)$$

and where the city-block norm $|\cdots|$ is defined by

$$|\mathbf{p}| \equiv \sum_i |p_i|. \qquad (4)$$

The system makes a *category choice* when at most one F_2 node can become active at a given time. The index J denotes the chosen category, where

$$T_J = \max\{T_j : j = 1, \ldots, N\}. \qquad (5)$$

If more than one T_j is maximal, the category with the smallest j index is chosen. In particular, nodes become committed in order $j = 1, 2, 3 \ldots$. When the Jth category is chosen, $y_J = 1$; and $y_j = 0$ for $j \neq J$. The $F_2 \rightarrow F_1$ signal vector $\mathbf{V}$ is then equal to the Jth category weight vector $\mathbf{w}_J$ and the F_1 activity vector $\mathbf{x}$ is reduced from $\mathbf{A}$ to the matched pattern $\mathbf{A} \wedge \mathbf{w}_J$. That is, in a choice system, the F_1 vector $\mathbf{x}$ obeys the equation

$$\mathbf{x} = \begin{cases} \mathbf{A} & \text{if } F_2 \text{ is inactive} \\ \mathbf{A} \wedge \mathbf{w}_J & \text{if the } J\text{th } F_2 \text{ node is chosen.} \end{cases} \qquad (6)$$

E. Resonance or Reset

Resonance occurs if the *match function* $|\mathbf{A} \wedge \mathbf{w}_J||\mathbf{A}|^{-1}$ of the chosen category meets the vigilance criterion

$$\frac{|\mathbf{A} \wedge \mathbf{w}_J|}{|\mathbf{A}|} \geq \rho \qquad (7)$$

that is, by (6), when the Jth category becomes active, resonance occurs if

$$|\mathbf{x}| = |\mathbf{A} \wedge \mathbf{w}_J| \geq \rho|\mathbf{A}|. \qquad (8)$$

Learning then ensues, as defined below. *Mismatch reset* occurs if

$$\frac{|\mathbf{A} \wedge \mathbf{w}_J|}{|\mathbf{A}|} < \rho \tag{9}$$

that is, if

$$|\mathbf{x}| = |\mathbf{A} \wedge \mathbf{w}_J| < \rho|\mathbf{A}|. \tag{10}$$

Then the value of the choice function T_J is set to 0 for the duration of the input presentation to prevent the persistent selection of the same category during search. A new index J represents the active category, selected by (5). The search process continues until the chosen J satisfies the matching criterion (7). By (1), search ends if J is an uncommitted node.

F. Learning

Once search ends, the weight vector $\mathbf{w}_J$ learns according to the equation

$$\mathbf{w}_J^{(\text{new})} = (1-\beta)\mathbf{w}_J^{(\text{old})} + \beta\big(\mathbf{A} \wedge \mathbf{w}_J^{(\text{old})}\big). \tag{11}$$

Fast learning corresponds to setting $\beta = 1$, when the weight vector $\mathbf{w}_J$ converges to the matched F_1 vector $\mathbf{x} = \mathbf{A} \wedge \mathbf{w}_J$ on each input presentation.

G. Normalization by Complement Coding

Normalization of fuzzy ART inputs prevents category proliferation as many weights erode to 0 in some input regimes. An $F_0 \rightarrow F_1$ input is normalized if $\sum_{i=1}^{2M} A_i = |\mathbf{A}| \equiv$ constant for all inputs $\mathbf{A}$. Complement coding automatically normalizes inputs because

$$|\mathbf{A}| = |(\mathbf{a}, \mathbf{a}^c)| = \sum_{i=1}^{M} a_i + \sum_{i=1}^{M}(1-a_i) = M. \tag{12}$$

IV. Fuzzy ART Geometry

A geometric interpretation of fuzzy ART represents each category as a box in M-dimensional space, where M is the number of components of input $\mathbf{a}$. In the prototype remote sensing example (Section VI), $\mathbf{a}$ represents two TM spectral band values for a given pixel, scaled to the interval $[0,1]$, so $M = 2$. With complement coding, then,

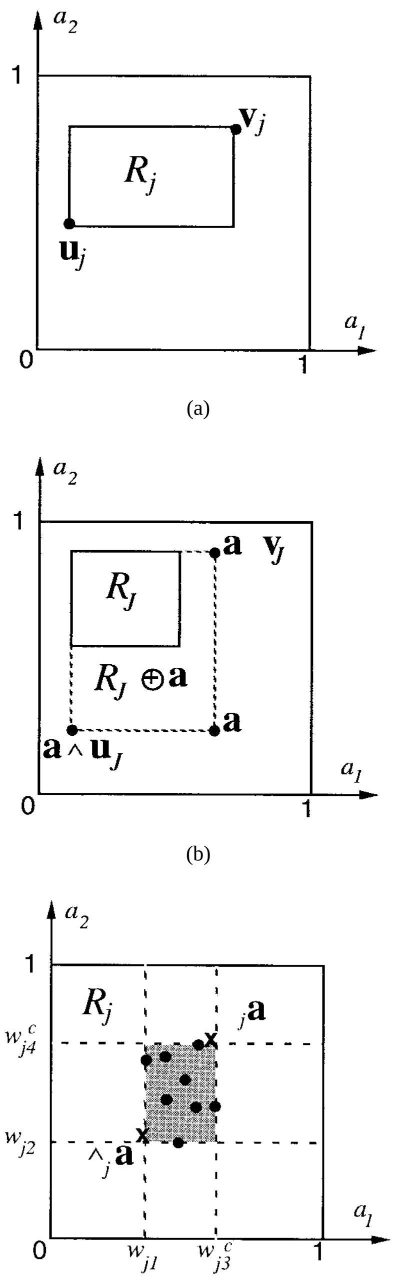
$$\mathbf{A} = (\mathbf{a}, \mathbf{a}^c) \equiv \big(a_1, a_2, a_1^c, a_2^c\big) \tag{13}$$

and each category j has a geometric representation as a rectangle R_j. Following the form of (13), a complement-coded weight vector $\mathbf{w}_j$ can be written as

$$\mathbf{w}_j = \big(\mathbf{u}_j, \mathbf{v}_j^c\big) \tag{14}$$

where $\mathbf{u}_j$ and $\mathbf{v}_j$ are two-dimensional (2-D) vectors. Vector $\mathbf{u}_j$ defines one corner of a rectangle R_j and $\mathbf{v}_j$ defines the opposite corner [Fig. 3(a)]. The size of R_j is

$$|R_j| \equiv |\mathbf{v}_j - \mathbf{u}_j| \tag{15}$$



(a)

(b)

(c)

Fig. 3. Fuzzy ART category boxes, with $M = 2$: (a) In complement coding form, each weight vector $\mathbf{w}_j$ has a geometric interpretation as a rectangle R_j with corners $(\mathbf{u}_j, \mathbf{v}_j)$. (b) During fast learning, R_J expands to $R_J \oplus \mathbf{a}$, the smallest rectangle that includes R_J and $\mathbf{a}$, provided that $|R_J \oplus \mathbf{a}| \leq 2(1-\rho)$. (c) With fuzzy ART fast learning and complement coding, the jth category rectangle R_j includes all those vectors $\mathbf{a}$ in the unit square that have activated category j without reset. The weight vector $\mathbf{w}_j$ equals $(\wedge_j \mathbf{a}, (\vee_j \mathbf{a})^c)$.

which is equal to the height plus the width of R_j. In the prototype example, each side of R_j represents a range of values of the corresponding TM band.

In a fast-learn fuzzy ART system, with $\beta = 1 (11), \mathbf{w}_J^{(\text{new})} = \mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$ when J is an uncommitted node. The corners of $R_J^{(\text{new})}$ are then $\mathbf{u}_J = \mathbf{a}$ and $\mathbf{v}_J = (\mathbf{a}^c)^c = \mathbf{a}$. Hence $R_J^{(\text{new})}$ is just the point box $\mathbf{a}$. Learning increases the size of R_J, which grows as weights shrink. Vigilance ρ determines the maximum size of R_J, with $|R_J| \leq M(1-\rho)$, as shown below. With fast learning, R_J expands to $R_J \oplus \mathbf{a}$, the minimum box containing R_J and $\mathbf{a}$ [Fig. 3(b)]. The corners of $R_J \oplus \mathbf{a}$ are $\mathbf{a} \wedge \mathbf{u}_J$ and $\mathbf{a} \vee \mathbf{v}_J$, where the fuzzy intersection $\wedge$ is defined by (3); and the fuzzy union $\vee$ is defined by

$$(\mathbf{p} \vee \mathbf{q})_i \equiv \max(p_i, q_i) \qquad (16)$$

[38]. Hence, by (15), the size of $R_J \oplus \mathbf{a}$ is

$$|R_J \oplus \mathbf{a}| = |(\mathbf{a} \vee \mathbf{v}_J) - (\mathbf{a} \wedge \mathbf{u}_J)|. \qquad (17)$$

However, before R_J can expand to include $\mathbf{a}$, category J is reset if $|R_J \oplus \mathbf{a}|$ would be too large, according to the vigilance criterion. With fast learning, R_j is the smallest box that encloses all vectors $\mathbf{a}$ that have chosen category j without reset.

If $\mathbf{a}$ has dimension M, the box R_j includes the two opposing vertices $\wedge_j \mathbf{a}$ and $\vee_j \mathbf{a}$, where the ith component of each of these vectors is

$$(\wedge_j \mathbf{a})_i = \min\{a_i : \mathbf{a} \text{ has been coded by category } j\} \qquad (18)$$

and

$$(\vee_j \mathbf{a})_i = \max\{a_i : \mathbf{a} \text{ has been coded by category } j\} \qquad (19)$$

[Fig. 3(c)]. The size of R_j is

$$|R_j| = |\vee_j \mathbf{a} - \wedge_j \mathbf{a}| \qquad (20)$$

and the weight vector $\mathbf{w}_j$ is

$$\mathbf{w}_j = (\wedge_j \mathbf{a}, (\vee_j \mathbf{a})^c) \qquad (21)$$

as in (14) and (15). Thus

$$|\mathbf{w}_j| = \sum_{i=1}^{M} (\wedge_j \mathbf{a})_i + \sum_{i=1}^{M} [1 - (\vee_j \mathbf{a})_i] = M - |\vee_j \mathbf{a} - \wedge_j \mathbf{a}| \qquad (22)$$

so the size of the box R_j is

$$|R_j| = M - |\mathbf{w}_j|. \qquad (23)$$

By (8), (11), and (12), the vigilance matching criterion implies a lower bound on the size of the weight vector $\mathbf{w}_j$

$$|\mathbf{w}_j| \geq \rho M. \qquad (24)$$

By (23) and (24)

$$|R_j| \leq (1 - \rho) M. \qquad (25)$$

Inequality (25) shows that high vigilance ($\rho \cong 1$) leads to small boxes R_j while low vigilance ($\rho \cong 0$) permits large R_j.

V. A Fuzzy ARTMAP Algorithm

ARTMAP networks for supervised learning self-organize mappings from input vectors, representing features such as patient history and test results, to output vectors, representing predictions such as the likelihood of an adverse outcome following an operation. The original binary ARTMAP [34] incorporates two ART 1 modules, ART_a and ART_b, that are linked by a *map field* F^{ab} (Fig. 2). At the map field the network forms associations between categories via outstar learning and triggers search, via the ARTMAP match tracking rule, when a training set input fails to make a correct prediction. Match tracking increases the ART_a vigilance parameter $\rho = \rho_a$ in response to a predictive error at ART_b. Fuzzy ARTMAP [35] substitutes fuzzy ART for ART 1.

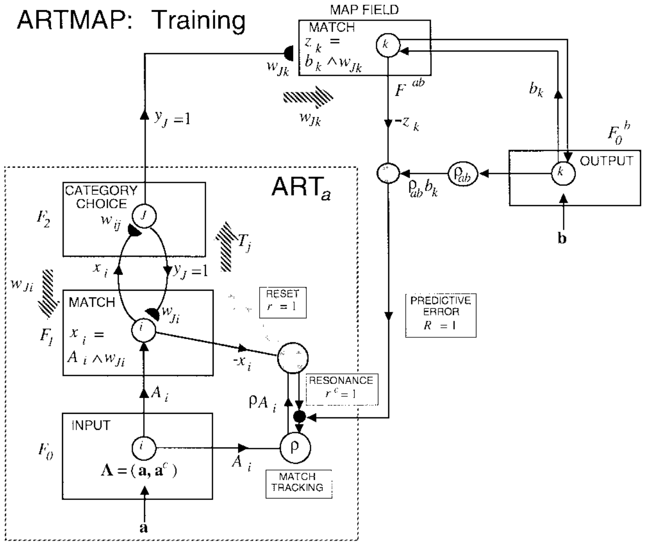


Fig. 4. A simplified ARTMAP network computes classification probabilities, with $|\mathbf{b}| = 1$ at an output field F_0^b.

Many applications of supervised learning systems such as ARTMAP are classification problems, where the trained system tries to predict a correct category given a test set input vector. A prediction might be a single category or distributed as a set of scores or probabilities. The fuzzy ARTMAP algorithm below outlines a procedure for applying fuzzy ART learning and prediction to this problem, which does not require the full ART_b architecture (Fig. 4). In the algorithm an input $\mathbf{a} = (a_1 \ldots a_i \ldots a_M)$ learns to predict an outcome $\mathbf{b} = (b_1 \ldots b_k \ldots b_L)$. A classification problem would set one component $b_K = 1$ during training, placing an input $\mathbf{a}$ in class K.

Note that the fuzzy ARTMAP algorithm allows a small match-tracking parameter (ε) to be either positive or negative. Compared to the original match tracking algorithm, which allowed only positive ε values $(MT+)$, a negative value of ε $(MT-)$ can facilitate prediction with sparse or inconsistent data and improve memory compression without loss of accuracy, and the resulting algorithm is actually a better approximation of the full ARTMAP differential equations [39].

A. Fuzzy ARTMAP Training

During training, input pairs $(\mathbf{a}^{(1)}, \mathbf{b}^{(1)})$, $(\mathbf{a}^{(2)}, \mathbf{b}^{(2)}), \ldots$, $(\mathbf{a}^{(n)}, \mathbf{b}^{(n)}), \ldots$ are presented for equal time intervals. Each ART_a input is complement coded, with $0 \leq a_i \leq 1$, $a_i^c \equiv 1 - a_i$, and $\mathbf{I} = \mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$, so that $|\mathbf{A}| \equiv M$. The output $\mathbf{b}$

is normalized to $1(\sum_{k=1}^{L} b_k = 1)$, corresponding to a category probability distribution. During testing, search may occur, if the baseline vigilance parameter ($\bar{\rho}$) is positive. Once a chosen F_2 node J meets the ART_a matching criterion, the predicted outcome probability distribution is the $F_2 \to F^{ab}$ weight vector $(w_{J1} \ldots w_{Jk} \ldots w_{JL})$, normalized to 1 at F_0^b.

1) Variables: $i = 1 \ldots 2M,\ j = 1 \ldots N,\ k = 1 \ldots L$

STM activation
x_i—F_1 (matching)
y_j—F_2 (coding)
z_k—F^{ab} (map field)

LTM weights
w_{ij}—$F_1 \leftrightarrow F_2$
w_{jk}—$F_2 \to F^{ab}$

$F_1 \to F_2$ signals
S_j—Phasic
Θ_j—Tonic
T_j—Total

C—#committed nodes
ρ—ART_a vigilance

2) Signal Rule: Define the $F_1 \to F_2$ signal function $T_j = g(S_j, \Theta_j)$, where $g(0,0) = 0$ and $\frac{\partial g}{\partial S_j} > \frac{\partial g}{\partial \Theta_j} > 0$ for $S_j > 0$ and $\Theta_j > 0$.
E.g.,

$$T_j = S_j + (1-\alpha)\Theta_j$$

with $\alpha \in (0,1)$ (choice-by-difference) or

$$T_j = S_j/(\alpha + 2M - \Theta_j)$$

with $\alpha > 0$ (Weber law).

The phasic signal component S_j equals $\sum_{i=1}^{2M} A_i \wedge w_{ij}$ and the tonic signal component Θ_j equals $\sum_{i=1}^{2M}(1 - w_{ij})$.

3) Notation
Minimum— $a \wedge b \equiv \min\{a, b\}$

4) Parameters
Number of input components—$i = 1 \ldots 2M$
Number of coding nodes—$j = 1 \ldots N$
Number of output components—$k = 1 \ldots L$
Signal rule parameters—E.g., $\alpha \in (0,1)$, (choice-by-difference) or $\alpha > 0$ (Weber law)
Learning rate—$\beta \in [0,1]$, with $\beta = 1$ for fast learning
Baseline vigilance (ART_a)—$\bar{\rho} \in [0,1]$, with $\bar{\rho} = 0$ for maximal code compression
Map field vigilance—$\rho_{ab} \in [0,1]$, with $\rho_{ab} \cong 1$ for maximal output separation
Match tracking—ε, with $|\varepsilon|$ small.

$$\mathrm{MT}+: \quad \varepsilon > 0$$
$$\mathrm{MT}-: \quad \varepsilon \le 0$$

F_2 order constants—$0 < \Phi_N < \ldots < \Phi_j < \ldots < \Phi_1 < g(M, 0)$, with all $\Phi_j \cong g(M, 0)$.

5) First Iteration: $n = 1$
$F_1 \leftrightarrow F_2$ weights—$w_{ij} = 1 \quad i = 1 \ldots 2M, \quad j = 1 \ldots N$
$F_2 \to F^{ab}$ weights—$w_{jk} = 1 \quad j = 1 \ldots N, \quad k = 1 \ldots L$
Number of committed nodes—$C = 0$
Signal to uncommitted nodes—$T_j = \Phi_j \quad j = 1 \ldots N$
ART_a vigilance—$\rho = \bar{\rho}$
Input—$A_i = \begin{cases} a_i^{(1)} & \text{if } 1 \le i \le M \\ 1 - a_i^{(1)} & \text{if } M+1 \le i \le 2M \end{cases}$
Output—$b_k = b_k^{(1)} \quad k = 1 \ldots L$

6) Reset: New STM steady state at F_2 and F_1
Choose a category—Let J be the index of the F_2 node with maximal input T_j, i.e.,

$$T_J = \max\{T_1 \ldots T_N\}$$

Number of committed nodes—If $J > C$, increase C by $1 (C = J)$
F_1 activation—$x_i = A_i \wedge w_{iJ} \quad i = 1 \ldots 2M$

7) MTM: $F_1 \to F_2$ signal is refractory on the time scale of search
$T_J = 0$

8) Reset or Prediction: Check the F_1 matching criterion
If $\sum_{i=1}^{2M} x_i < \rho M$, go to 6) Reset
If $\sum_{i=1}^{2M} x_i \ge \rho M$, go to 9) Prediction

9) Prediction:
F^{ab} activation—$z_k = b_k \wedge w_{Jk} \quad k = 1 \ldots L$

10) Match tracking or resonance: Check the F^{ab} matching criterion
If $\sum_{k=1}^{L} z_k < \rho_{ab}$, go to 11) Match tracking
If $\sum_{k=1}^{L} z_k \ge \rho_{ab}$, go to 12) Resonance

11) Match tracking: Raise ρ to the point of ART_a reset
$\rho = \frac{1}{M} \sum_{i=1}^{2M} x_i + \varepsilon$
Go to 6) Reset

12) Resonance: New LTM weights on the time scale of learning
Old weights—$w_{iJ}^{\mathrm{old}} = w_{iJ} \quad i = 1 \ldots 2M,$

$$w_{Jk}^{\mathrm{old}} = w_{Jk} \quad k = 1 \ldots L$$

Decrease $F_1 \leftrightarrow F_2$ weights—$w_{iJ} =$

$(1-\beta) w_{iJ}^{\mathrm{old}} + \beta(A_i \wedge w_{iJ}^{\mathrm{old}}) \quad i = 1 \ldots 2M$
Decrease $F_2 \to F^{ab}$ weights—$w_{Jk} =$

$(1-\beta) w_{Jk}^{\mathrm{old}} + \beta(b_k \wedge w_{Jk}^{\mathrm{old}}) \quad k = 1 \ldots L$
ART_a vigilance recovery—$\rho = \bar{\rho}$

13) Next iteration:
Increase n by 1
New input—$A_i = \begin{cases} a_i^{(n)} & \text{if } 1 \le i \le M \\ 1 - a_i^{(n)} & \text{if } M+1 \le i \le 2M \end{cases}$
New output—$b_k = b_k^{(n)} \quad k = 1 \ldots L$
New F_1 activation—$x_i = A_i \wedge w_{iJ} \quad i = 1 \ldots 2M$
New $F_1 \to F_2$ signal to committed nodes

Phasic—

$$S_j = \sum_{i=1}^{2M} A_i \wedge w_{ij} \quad j = 1 \ldots C$$

Tonic—

$$\Theta_j = \sum_{i=1}^{2M} (1 - w_{ij}) \quad j = 1 \ldots C$$

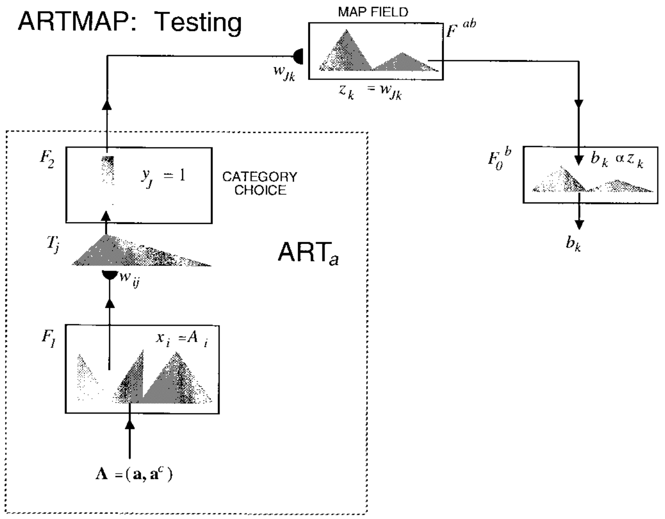


Fig. 5. During ARTMAP testing, an input $\mathbf{a}$ activates the Jth F_2 category node. The map field weights w_{Jk} then form a prediction vector $\mathbf{z}$, which may be distributed. The network computes classification probabilities, with $|\mathbf{b}| = 1$, at the output field F_0^b.

Total—

$$T_j = g(S_j, \Theta_j) \quad j = 1 \ldots C \quad \text{(2) Signal rule)}$$

Go to (6) Reset

B. Fuzzy ARTMAP Testing

During ARTMAP testing (Fig. 5), $F_1 \leftrightarrow F_2$ categorization weights w_{ij} and $F_2 \rightarrow F^{ab}$ prediction weights w_{jk} are fixed. A test-set input $\mathbf{a}$ chooses an ART_a category J, possibly following search, if $\bar{\rho} > 0$. Map field activation $\mathbf{z}$ then equals the $F_2 \rightarrow F^{ab}$ weight vector $(w_{J1} \ldots w_{Jk} \ldots w_{JL})$, and the output vector $\mathbf{b}$ equals this vector normalized to 1. With fast learning when $\mathbf{b}$ represents single output classes during training, only one weight w_{JK} is positive and only one component of $\mathbf{b}$ is positive, corresponding to prediction of the single output class $k = K$. If $\mathbf{b}$ is distributed during training or if learning is slow, $\mathbf{b}$ may represent a probability vector, distributed across output classes.

ARTMAP fast learning typically leads to different adaptive weights and recognition categories for different orderings of a given training set, even when the overall predictive accuracy of each such trained network is similar. The different category structures cause the location of test set inputs where errors occur to vary as the training set input orderings vary. A voting strategy uses several ARTMAP systems that are separately trained on one input set with different orderings. The final prediction for a given test set item is the one made by the largest number of networks in a voting "committee." Since the set of items making erroneous predictions varies from one ordering to the next, voting serves both to cancel many of the errors and to assign confidence estimates to competing predictions. A committee of about five voters has proved suitable in many examples, and the marginal benefits of voting are most apparent when the number of training samples is limited.

For voting, ARTMAP generates a set of prediction vectors for each of the trained networks produced by several different orderings of the training set inputs. The voting networks may average their output vectors $\mathbf{b}$ for each input $\mathbf{a}$; or each voting network may choose one output class, with the predicted class being the one that receives the most votes.

1) Test set input:

Input— $A_i = \begin{cases} a_i & \text{if } 1 \le i \le M \\ 1 - a_i & \text{if } M+1 \le i \le 2M \end{cases}$

2) $F_1 \rightarrow F_2$ signal:

Phasic—

$$S_j = \sum_{i=1}^{2M} A_i \wedge w_{ij} \quad j = 1 \ldots C$$

Tonic—

$$\Theta_j = \sum_{i=1}^{2M} (1 - w_{ij}) \quad j = 1 \ldots C$$

Total—

$$T_j = \begin{cases} g(S_j, \Theta_j) & j = 1 \ldots C \text{ (Signal rule)} \\ \Phi_j & j = C+1 \ldots N \end{cases}$$

3) F_2 category choice:

Let J be the index of the F_2 node with maximal input T_j, i.e.,

$$T_J = \max\{T_1 \ldots T_N\}$$

4) Output prediction:

$$b_k = \frac{w_{Jk}}{\sum_{\kappa=1}^{L} w_{J\kappa}} \quad k = 1 \ldots L$$

VI. Remote Sensing Prototype Example

A simplified remote sensing classification problem illustrates fuzzy ARTMAP dynamics and also serves as a prototype for the remote sensing tests described in Section VIII. The prototype task is learning to identify one of three CALVEG [40] vegetation classes (mixed conifer, coast live oak, southern mixed chaparral) for sites at which two spectral values (Landsat TM 1 and 4) are known at each pixel. The prototype example is based on a data set collected at the Cleveland National Forest. Larger scale tests on this data set (Section VIII) predict 8 possible vegetation classes with inputs of up to 6 TM bands and 7 ancillary variables. In this more realistic setting, fuzzy ARTMAP performance is compared with that of maximum likelihood [41], [42], K Nearest Neighbor [43], and back propagation [21], [22]. However, first reducing the number of input dimensions to two (TM bands) and the number of output classes to three (vegetation classes) will allow visual illustration of fuzzy ARTMAP dynamics (Section VII).

The data set for the prototype remote sensing problem reports the vegetation class for each of 50 sites: 16 mixed conifer, 25 coast live oak, and nine southern mixed chaparral [Table I(A)]. The sites vary in size, averaging about 90 pixels each. Landsat spectral bands TM1 and TM4 constitute the

TABLE I
PROTOTYPE REMOTE SENSING TESTS

A. Data set

Class label	# sites	# pixels
mixed conifer	16	1336
coast live oak	25	2752
southern mixed chaparral	9	348
TOTAL	50	4436

B. Fuzzy ARTMAP Incremental Learning

Training set (# pixels)	Categories (# F_2 nodes)	Test set pixels (% correct)	Test set sites (# correct)
100	8	85.9%	8/10
500	21	83.2%	9/10
2000	72	88.5%	10/10
3328	126	89.3%	10/10

C. Voting

Input ordering (Figure 8)	Categories (# F_2 nodes)	Test set pixels (% correct)	Test set sites (# correct)
(a)	126	89.3%	10/10
(b)	131	86.8%	9/10
(c)	139	86.8%	9/10
(d)	153	89.4%	9/10
(e)	133	84.8%	8/10
average	136	87.4%	9/10
voting	---	91.0%	10/10

data set input for each pixel, with values scaled to the interval $[0, 1]$. Before training, 10 sites, representative of the vegetation class mix, are reserved as a test set. No pixels from these sites are used during training. The goal is to predict the correct vegetation class label for each of the 10 test set sites.

During training and testing, a given pixel corresponds to an ART_a input $\mathbf{a} \equiv (a_1, a_2)$, where a_1 is the value of TM1 and a_2 is the value of TM4 at that pixel. The corresponding ART_b input vector $\mathbf{b}$ represents the CALVEG vegetation class of the pixel's site

$$\mathbf{b} = \begin{cases} (1,0,0) & \text{mixed conifer} \\ (0,1,0) & \text{coast live oak} \\ (0,0,1) & \text{southern mixed chaparral.} \end{cases} \tag{26}$$

During training, vector $\mathbf{b}$ informs the ARTMAP network of the vegetation class to which the pixel's site belongs. This supervised learning process allows adaptive weights to encode the correct association between $\mathbf{a}$ and $\mathbf{b}$. Tests below examine the effect of training set size on predictive accuracy [Table I(B)]. To generate a training set of a given size, pixels are selected at random from the entire training set, which represents approximately 3600 pixels in 40 sites. Other tests show how voting can improve predictive accuracy [Table I(C)].

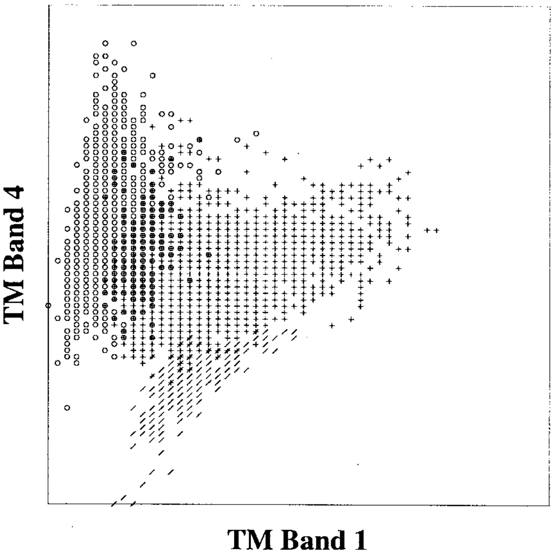


Fig. 6. Prototype remote sensing inputs. Each point shows the scaled Landsat spectral band components a_1 (TM1—blue) and a_2 (TM4—near infrared) of the ART_a input vector $\mathbf{a}$. Points o are found in mixed conifer sites, points + are found in coast live oak sites, and points / are found in southern mixed chaparral sites. Data set values are taken from the Cleveland National Forest.

During testing, each test set pixel predicts a class, given the spectral band input values a_1 and a_2 for that pixel. Performance accuracy is measured both in terms of the percent of pixels that are correct and in terms of the fraction of sites that are correctly identified by a vote among pixels in the site.

VII. FUZZY ARTMAP PROTOTYPE TESTS

The prototype remote sensing tests illustrate fuzzy ARTMAP dynamics by showing how the network learns to make correct vegetation class predictions. Fig. 6 indicates why the problem is difficult: of the 4436 pixels in the data set [Table I(A)], many share spectral band values within and between the three vegetation classes, and the three classes are not linearly separable. In fact the problem proved to be too difficult for an elementary back propagation network to make accurate predictions (Section D).

During the initial learning phase, pixels are selected one at a time, at random, from the 40 training set sites. Fuzzy ARTMAP is trained incrementally, with each TM band vector a presented just once. Following a search, if necessary, the network selects an ART_a category by activating an F_2^a node J for the input pixel, then learns to associate category J with the ART_b vegetation class of the site in which the pixel is located. With fast learning, the class prediction of each ART_a category J is permanent. If some input $\mathbf{a}$ with a different class prediction later selects this category, match tracking will raise ART_a vigilance ρ just enough to trigger a search for a different ART_a category. In all prototype tests, $\alpha \cong 0$ (conservative limit—Section II-A), $\beta = 1$ (fast learning—Section III-F), and $\bar{\rho} = 0$ (maximal code compression). The map field vigilance ρ_{ab} can have an arbitrary value between 0 and 1, since with fast learning and binary predictions the map field registers

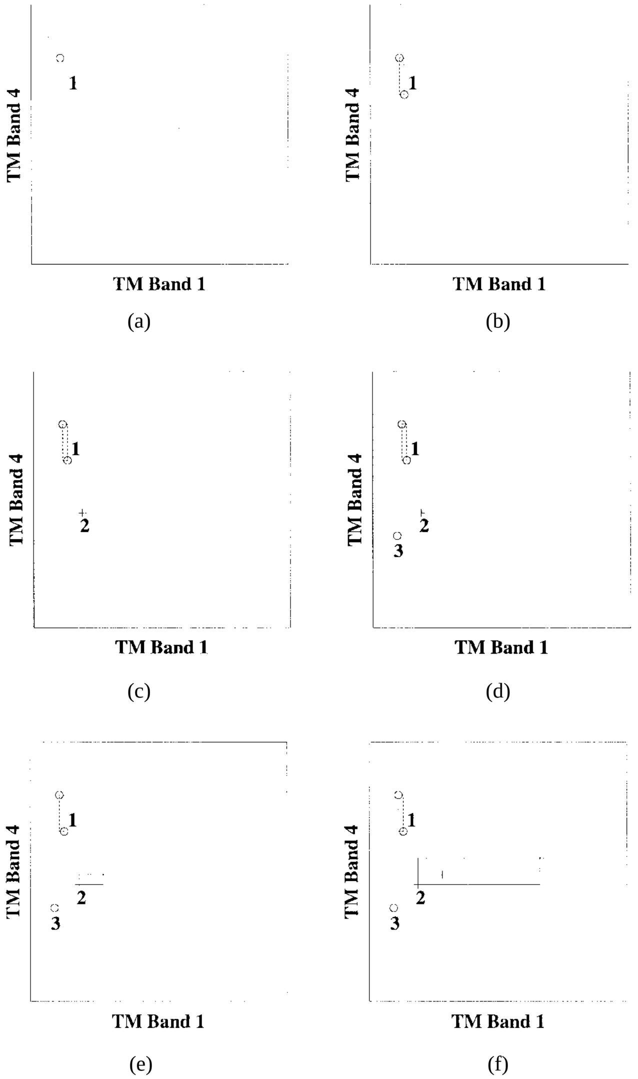


Fig. 7. Prototype remote sensing example: Fuzzy ARTMAP incremental learning in response to the first six training set points. Inputs 1(a), 2(b), and 4(d) are from mixed conifer sites (o) and inputs 3(c), 5(e), and 6(f) are from coast live oak sites $(+)$. After learning, inputs 1 and 2 have established the ART_a category $J = 1$, which maps to mixed conifer; inputs 3, 5, and 6 have established category $J = 2$, which maps to coast live oak; and input 4 has established the point category $J = 3$, which maps to mixed conifer. Southern mixed chaparral, with sites that include less than 8% of the pixels, happened not to be represented among the first six inputs, which were selected at random.

either a perfect match $(|\mathbf{z}| = 1)$ or a complete mismatch $(|\mathbf{z}| = 0)$.

A. Incremental Learning by the First Six Inputs

Fig. 7 illustrates fuzzy ARTMAP learning in response to the first six training set inputs, selected at random from the 40 training set sites. Input 1 [Fig. 7(a)] represents a pixel that has a low TM1 value (a_1) and a high TM4 value (a_2) and that is found at a mixed conifer site $(\circ)$. Input $\mathbf{a}$ selects the uncommitted F_2 node $J = 1$ (Section III-B). During learning, all weights w_{Jk} from this node to the map field F^{ab} (Fig. 2) decay to 0 except for the weight w_{Jk} to the node K representing the correct vegetation class $(K = 1)$. Category $J = 1$ appears as the point box R_1.

Input 2 [Fig. 7(b)] also selects category $J = 1$. At the start of each input presentation, the ART_a vigilance ρ equals the baseline vigilance $\bar{\rho}$, which here equals 0. Therefore, $\mathbf{a}$ meets the ART_a matching criterion (Section III-E), so category $J = 1$ remains active and predicts, via the map field, that this new input is also from a mixed conifer site. Since this prediction is correct, field F^{ab} registers a perfect match $(|\mathbf{z}| = 1)$ and so meets the map field matching criterion. During learning the category box R_1 expands to include input point 2.

Input 3, from a coast live oak site $(+)$, requires match tracking and search to learn the correct prediction, as follows [Fig. 7(c)]. This input $\mathbf{a}$ first selects category $J = 1$. Again, since $\rho = \bar{\rho} = 0$, ART_a accepts the new input into this category long enough to predict mixed conifer. However, the network now detects a predictive error, since the incorrect prediction sends the activity z_k of all map field nodes to 0. Match tracking increases ρ just enough to reset ART_a, where a new node $J = 2$ becomes active. Since uncommitted nodes meet the matching criterion for any ρ, node $J = 2$ remains active, establishing the point box R_2, which henceforth will predict coast live oak $(K = 2)$.

Input 4, again from a mixed conifer $(\circ)$ site, shows how match tracking can create more than one box for each class. This feature allows ARTMAP to learn a set of decision rules of arbitrary complexity while minimizing predictive error. For example, concentric rings in an input space could be mapped to alternating category predictions. At the same time, setting $\bar{\rho}$ equal to 0 allows the network to maximize code compression, creating a new category only in response to a predictive error. Design principles that balance the two goals—minimum error, maximum compression—allow ARTMAP to learn correct predictions for a small category of rare cases embedded in a large category of common cases. Input 4 [Fig. 7(d)] first selects the F_2 point category $J = 2$, which maximizes the choice function T_j (2). Since this category predicts coast live oak, the map field registers a mismatch, which sends a match tracking signal to ART_a. This raises ρ until it is just above the match ratio $|\mathbf{A} \wedge \mathbf{w}_J||\mathbf{A}|^{-1}$ (Section V), where $\mathbf{A} \equiv (\mathbf{a}, \mathbf{a}^c)$ is the complement coded input to F_1 (Section III-A). The next category J that will be able to resonate, and so remain active long enough to make a class prediction, must now meet the stricter matching criterion imposed by the new, higher ART_a vigilance ρ. Geometrically (Section IV), once node $J = 2$ leads to match tracking, a new active category J will now meet the ART_a matching criterion only if the expanded box $\mathrm{R}_J \oplus \mathbf{a}$ would be *smaller* than $R_2 \oplus \mathbf{a}$, where $\mathbf{a}$ is the current input. After match tracking, input 4 next selects category $J = 1$ (which actually would have made the correct prediction), but this category fails to meet the ART_a matching criterion, since the box $R_1 \oplus \mathbf{a}$ would have been larger than $R_2 \oplus \mathbf{a}$. The input therefore also resets node $J = 1$. then activates the uncommitted node $J = 3$, which learns to predict mixed conifer $(K = 1)$.

Input 5 [Fig. 7(e)] selects category node $J = 2$, which correctly predicts coast live oak $(+)$, so no match tracking or ART_a search is invoked. During learning, as the weight vector $\mathbf{w}_2$ adapts according to (11), the box R_2 expands to $R_2 \oplus \mathbf{a}$, where $\mathbf{a}$ represents the TM values of input 5. Since

$\rho = \bar{\rho} = 0$, the size of $R_2 \oplus \mathbf{a}$ is unrestricted. Finally, input 6 [Fig. 7(f)] selects and further expands box R_2. Weights remain unchanged during learning only if $\mathbf{a}$ is inside a selected box that has already learned to make the correct prediction. As training proceeds, category boxes cover more of the input space, so the case where weights remain unchanged during learning occurs increasingly often. If a finite input set is presented repeatedly, all training set inputs learn to predict with 100% accuracy, provided that the set of input predictions is consistent, i.e., that no two identical inputs $\mathbf{a}$ make the same vegetation class prediction.

B. Predictions of the Trained ARTMAP Network

As incremental learning proceeds, fuzzy ARTMAP creates a set of overlapping category boxes R_j, each predicting one of the three vegetation classes. By the time 100 training set pixel inputs have been selected at random from the 40 training set sites, fuzzy ARTMAP has created eight categories [Table I(B)]. Three of these categories predict mixed conifer, four predict coast live oak, and one predicts southern mixed chaparral. The 10 test set sites contain a total of 1108 pixels. After training on the first 100 inputs, network performance at this stage of learning was first measured by the number of correct vegetation class predictions the test set pixels were able to make. For each test set pixel, the TM band vector $\mathbf{a}$ selects one of the eight ART$_a$ categories, then predicts that its site belongs to the vegetation class associated with that category. After training on just 100 input points, 85.9% of the test set pixels correctly predicted the vegetation classes of their sites. A second performance measure examined the number of test set sites that would be correctly classified. This method counts the number of pixels in each site that predict each vegetation class, then selects the class chosen by the most pixels. At this stage of learning, having used only 3% of the training set pixels, eight of the 10 test site vegetation classes were correctly identified. In this case, too few southern mixed chaparral exemplars had been presented for that class to easily win a majority at any site.

As the number of training set inputs increased, the pixel-level predictive accuracy increased only marginally, even decreasing transiently as the number of training set inputs increased from 100 to 500 [Table I(B)]. After presentation of all 3328 training set pixels, 89.3% of the test set pixels correctly predict the vegetation class of their site. However, site-level prediction improves steadily to 9/10 test set sites, after training on 500 inputs; and 10/10 sites, after training on 2000 inputs or on the full training set. This result highlights the observation that the pixel is often too small and noisy a unit to make an accurate prediction. However, a group of noisy pixel-level results can be pooled to form accurate mappings across functional regions or sites.

C. Voting

A typical characteristic of fast learning is dependence of category structure upon the order of training set input presentation. For example, suppose that two fuzzy ARTMAP networks learn from a common input set that is presented in two different orders during training. The two networks might then each correctly predict 90% of the test set inputs, despite the fact that the two have significantly different internal input grouping rules, or category boxes, at ART_a. In particular, the test set inputs that the first network identifies correctly are typically different from those that the second network identifies correctly, despite the fact that both were trained on the same input set with the same network parameters. ARTMAP voting uses this order dependence to advantage to improve and stabilize overall predictive performance, as follows.

Fig. 8(a)–(e) illustrates the decision regions of the prototype remote sensing example after presentation of all 3328 training set inputs (Table 1-C). A decision region plot shows predictions all TM band inputs $\mathbf{a}$ would make if presented to the trained network. In Fig. 6, data set points from mixed conifer sites are represented by a circle $(\circ)$, points from coast live oak sites by a plus $(+)$, and points from southern mixed chaparral sites by a slash $(/)$. The same marks indicate vegetation class predictions made by a network in response to spectral value inputs across the unit square. The rough decision boundaries reflect the ambiguous predictions in the corresponding portion of the data set.

Fig. 8(a)–(e) and Table I(C) show how network predictions can vary as a function of input order. Each of these five tests uses the same training set, presented in different, randomly chosen, orders. Decision boundaries vary, as do the number of ART_a categories (from 126 to 153), the number of correct test set pixels (from 84.8% to 89.4%), and the number of correct test set site identifications (from 8/10 to 10/10). Before knowing the test set answers, it would be difficult to decide which of these five networks would be the most accurate on novel data. ARTMAP voting chooses for each pixel the class prediction chosen by the largest number of the five “voting committee” networks. The size of each vote also provides a measure of confidence in each decision. Confidence is typically lowest near decision boundaries. Fig. 8(f) indicates how voting can smooth and stabilize decision boundaries. In addition, pixel-level performance on the voting network (91.0%) is better than that of any individual trained network, and site-level prediction is perfect (10/10).

D. Back Propagation Tests

An elementary back propagation neural network did not perform well on the prototype remote sensing problem. Networks were trained using a variety of parameters, initial conditions, and numbers of hidden units. In all cases, back propagation was unable to make correct predictions for the southern mixed chaparral sites. The most successful network used 10 hidden units, a learning rate of 0.3, and a momentum rate of 0.4. After presentation of the full training set, this system correctly identified the vegetation class of just 75% of the test set pixels. Site-level prediction was correct for the seven mixed conifer and coast live oak training set sites. However, all three southern mixed chaparral sites were wrongly identified, evidently because these “rare cases” were averaged away among the more common exemplars. On the other hand,

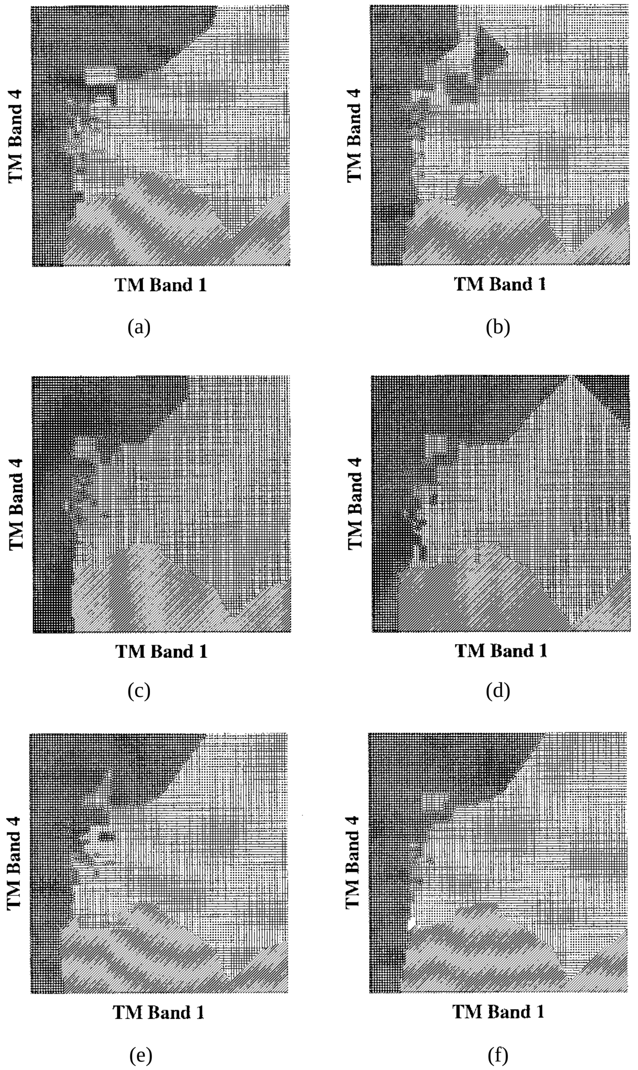


Fig. 8. Prototype remote sensing example: Fuzzy ARTMAP voting. (a)–(e) Fuzzy ARTMAP networks trained on a common set of 3328 inputs presented in five different, random orders show variations in decision region geometry. Points marked by a circle (○) predict mixed conifer, points marked by a plus (+) predict coast live oak, and points marked by a slash (/) predict southern mixed chaparral. Pixel-level predictive accuracy ranges from (e) 84.8% to (d) 89.4% while site-level predictive accuracy ranges from (e) 8/10 to (a) [Table I(C)] 10/10. (f) Voting across the five trained networks boosts pixel-level accuracy to 91.0% and site-level accuracy to 10/10. Blank spaces indicate a 2-2-1 tie among the voters.

sophisticated variations of the back propagation algorithm that have recently been developed might improve performance.

VIII. Remote Sensing Tests

Tests in this section show how fuzzy ARTMAP performance on the Cleveland Forest data set compares to that of standard maximum likelihood methods [42]. Voting improves ARTMAP predictive success, and both systems benefit from appropriate selection of input variables and predictive confidence thresholds. Best results are obtained by a hybrid system based on a convex combination of fuzzy ARTMAP and maximum likelihood predictions. As in the ARTMAP voting process (Section VII-C), hybrid prediction takes advantage of the fact that fuzzy ARTMAP and maximum likelihood tend to make errors under somewhat different circumstances.

TABLE II
Remote Sensing Data Set

	CALVEG class	# sites	# pixels
	mixed conifer	16	1336
	canyon live oak	11	814
	coast live oak	25	2752
	chamise	21	1042
	scrub oak	20	1315
	red shanks	11	1450
	southern mixed chaparral	9	348
	northern mixed chaparral	50	2398
TOTAL:	**8 classes**	**163 sites**	**11,455 pixels**

A. Cleveland National Forest Test Stand Data

The test stands from the Cleveland National Forest identify the CALVEG vegetation class for 209 sites. The full data set represents 17 vegetation classes. The primary goal of this study was to develop and compare automated classification methods for large-scale remote sensing applications. In order to focus on the methods, the selected prediction problem could not be too easy, but neither could it be dominated by noise or chance. The test data set examined here thus excludes vegetation classes represented by only a few sites, leaving eight vegetation classes and 163 sites (Table II). The prediction problem remains challenging and realistic: the pixel-based (25 × 25 m) remotely sensed data are typically noisy and unreliable; the number of training set sites (143) is small relative to the number of classes (eight); some of the vegetation classes, such as the three different types of oak, are likely to have similar features; and the actual vegetation at each site, where sites range in size from 9 to 610 pixels (5625–381 250 m^2), is, in all likelihood, not a pure sample of just one class.

B. Input Variable Combinations

For each pixel, the Cleveland Forest data set provides six Landsat Thematic Mapper (TM) band values, three linear combinations of the TM band values, and four terrain variables. The three linear combinations of TM1-5&7 reflect brightness (B), greenness (G), and wetness (W) [44]. The four terrain variables—slope (SL), aspect (A), shade (SH), and elevation (E)—were derived from digital elevation models, warped to fit the Landsat image [16], [20], [45].

Tests in this section focus primarily on fuzzy ARTMAP and maximum likelihood performance on data sets for which input $\mathbf{a}$ provides only the six TM values (combo 1) and on data sets for which $\mathbf{a}$ provides all 13 input variables (combo 2) (Table III). On tests that use each of these input variable combinations, basic fuzzy ARTMAP and maximum

TABLE III
TEST RESULTS

A. Basic maximum likelihood and fuzzy ARTMAP

Combo #	Input variables	% Correct maximum likelihood	% Correct + (# F_2 nodes) fuzzy ARTMAP
1	TM1-5&7	46	44.7 (1203 cats)
2	TM1-5&7 B, G, W SL, A, SH, E	54	57.2 (208 cats)
3	B, G, W	40	48.1 (1145 cats)
4	TM1-5&7 SL, A, SH, E	54	47.2 (365 cats)
5	TM1-5&7 SL, A, E	54	48.5 (392 cats)
6	TM1-5&7 B, G, W	46	51.9 (595 cats)
7	B, G, W SL, A, SH, E	52	56.6 (259 cats)

B. K Nearest Neighbor (KNN)

Combo #	Inputs	K=1	K=5	K=10
1	TM1-5&7	44.3%	47.0%	44.0%
2	TM1-5&7 B, G, W SL, A, SH, E	56.0%	54.0%	56.0%

C. Confidence thresholds and site-level voting

Combo 1 (6 variables)

Maximum likelihood		Fuzzy ARTMAP		
CT=-∞	CT=-21.6	$\bar{\rho}$=0	$\bar{\rho}$=0.87	$\bar{\rho}$=0.87, 5 voters
46.0%	48.8%	44.7%	46.4%	48.6%

Combo 2 (13 variables)

Maximum Likelihood		Fuzzy ARTMAP	
CT=-∞	CT=10.0	$\bar{\rho}$=0	$\bar{\rho}$=0, 5 voters
54.0%	56.5%	57.2%	60.0%

likelihood [Table III(A)] have similar predictive accuracies: approximately 45–46% with the six TM input variables and 54–57% with all 13 variables. On data sets that provide various subsets of the 13 input variables, performance of the two systems can differ significantly. For example, when pixel inputs provide only the three linear combinations B, G, W (combo 3), maximum likelihood performance drops to 40% while fuzzy ARTMAP performance increases to 48%.

The patterns in the results of maximum likelihood classification as a function of inputs are consistent with past experience in remote sensing. Raw spectral bands have frequently produced better results than transforms such as brightness, greenness, and wetness. Similarly, combining the linearly transformed variables brightness, greenness, and wetness with the original spectral bands yields no improvement (combo 6).

The results of fuzzy ARTMAP classification are strikingly different, with the brightness, greenness, and wetness transforms resulting in better performance than the original spectral bands (combo 3). Even more divergent from maximum likelihood is the improved performance when fuzzy ARTMAP uses both the six spectral bands and the three linear transforms of the spectral band variables. In fact, one of the most interesting results of these tests is the increase in fuzzy ARTMAP performance from 44.7% to 51.9% when linear transforms are combined with the original spectral band inputs (combo 6). This result is in direct contrast with statistically-based classifiers. It also emphasizes the importance of selection of input features and suggests that performance might be further enhanced by other unknown transforms. On the other hand, ancillary variables have similar effects on maximum likelihood and fuzzy ARTMAP performance.

The first tests described here use a basic fuzzy ARTMAP network to predict the eight vegetation classes. Test procedures are like those of the prototype problem (Section VII-B), except that the prototype used only two spectral band values to predict three of the vegetation classes. As in the prototype tests, baseline vigilance $\rho = 0$ (maximal compression), $\alpha \cong 0$ (conservative limit), and $\beta = 1$ (fast learning), and there is only one voter. Learning is incremental, with each input presented once. During testing, classification accuracy is measured by site, with a site's vegetation class predicted to be the one chosen by the largest number of pixels. For sites at which ties occur, the number of correct classifications is counted at chance. In each test, the training set data represent 143 sites, with the remaining 20 sites providing the test set. In order to check for sampling bias in the test set selection, five different tests sets, each with 20 sites, were compared across multiple tests, with fuzzy ARTMAP and maximum likelihood using the same training and test sets. In addition, fuzzy ARTMAP was run with 35 different orderings of each training set, since input order could affect results by 1–2%.

C. K Nearest Neighbor and Back Propagation Tests

The K Nearest Neighbor (KNN) algorithm [43] was also tested on the six variable and 13 variable input sets [Table III(B)]. Predictive accuracy was similar to that of fuzzy ARTMAP and maximum likelihood, varying somewhat with the number of neighbors (K) chosen during testing. However, KNN needs to store all training set pixel vectors (approximately 10 000), while fuzzy ARTMAP compresses memory by a factor of 8 for combo 1, creating about 1200 ART_a categories during learning. Remarkably, using all 13 input variables, the average number of ART_a categories drops to 208, giving a compression ratio of 48:1 compared to KNN.

Although the back propagation neural network has been applied successfully to remote sensing classification problems (e.g., [46]), performance of an elementary back propagation system was not satisfactory on the present remote sensing problem. On combo 1, with TM1-5&7 as inputs, correct prediction rates ranged from 22% to 46% as the number of hidden units ranged from 15 to 60. The best test set prediction rate, obtained using 30 hidden units, was comparable to the average performance rates of maximum likelihood, KNN, and fuzzy ARTMAP. For this test, back propagation had a learning rate of 0.3 and momentum equal to 0.4, and each case was repeated 5 times, varying the set of initial weights. On the 13-variable input set (combo 2), the best back propagation performance was worse than the average performance of ARTMAP, maximum likelihood, and KNN: with 50 hidden units, a learning rate of 0.6, and momentum equal to 0.4, performance accuracy reached a maximum of 47.1% at the pixel level and 50% at the site level. Even with over 100 000 input presentations and 212 min of CPU time on a Sun 4 Sparc Station, weights did not converge during training. At lower learning rates, with CPU times exceeding 1000 min, back propagation's predictive accuracy was less than 27%. In general, back propagation requires slow learning and many presentations of each input, while fuzzy ARTMAP learning is fast and incremental, or "on-line." In addition, choosing the number of hidden units and optimizing the architecture typically require extensive simulation studies. Fuzzy ARTMAP is thus particularly well suited to ongoing training in situations where new information continues to arrive during use. Again, however, sophisticated application of a back propagation variant may have shown better performance.

D. Rejecting Low-Confidence Predictions

In the tests described in Section B, basic fuzzy ARTMAP and maximum likelihood systems make a vegetation class prediction for each pixel input, even if there is no good match between an input and any learned class. Performance accuracy of both classifiers can be boosted by adding a confidence threshold. Then, when confidence in a prediction is low, the corresponding test set pixel is labeled "inconclusive" and does not participate in the site-level vegetation class decision.

For fuzzy ARTMAP the matching criterion imposed by the baseline vigilance $\bar{\rho}$ provides a natural confidence threshold (Section V). During training, some category boxes R_j may grow large, since $\bar{\rho}$ is set equal to 0 in order to maximize code compression. During testing, ρ remains equal to $\bar{\rho}$. Setting $\bar{\rho} > 0$ imposes a minimum matching criterion before a chosen category J is allowed to remain active and make a prediction. Geometrically, a prediction requires that the size of the expanded box $R_J \oplus \mathbf{a}$ would be no greater than $(1-\bar{\rho})M$ (25), where M is the number of input variables. In particular, if R_J is already larger than then $(1-\bar{\rho})M$ any pixel that first chooses category J during testing will be regarded as inconclusive. Maximum likelihood computes a discriminant function value (DFV) for each vegetation class, given a pixel input. A confidence threshold (CT) checks that the maximal DFV of a pixel is greater than CT before that pixel may participate in a site-level prediction.

Choosing an optimal confidence threshold for a given test set would require prior knowledge of the test set inputs. However, a useful range of $\bar{\rho}$ (fuzzy ARTMAP) or CT (maximum likelihood) values can be estimated by reserving a randomly chosen portion of the training set as a "verification set." Before training on these inputs, tests would estimate a predictive confidence threshold that gives good performance on this subset. During testing, then, the threshold would be fixed at the selected value. The verification set procedure was used to obtain $\bar{\rho}$ values. The same method could have been used to select maximum likelihood CT values. However, the CT was here chosen simply to optimize maximum likelihood test set performance.

Fuzzy ARTMAP results were found to be fairly constant across wide $\bar{\rho}$ intervals. Moderate threshold levels boosted performance somewhat when training produced many categories, as in the six variable tests (combo 1), which average 1203 categories [Table III(C)]. As the threshold increases, at some point performance tends to increase for a short interval, then drop steeply, when the threshold is set so high that many useful predictions are discarded. With both combo 1 and combo 2, maximum likelihood performance shows a similar trend as the confidence threshold increases. In contrast, on the 13-variable input set (combo 2), where fuzzy ARTMAP produces only 208 categories, setting $\bar{\rho} = 0$ gives optimal performance, and performance begins to drop significantly for $\bar{\rho} > 0.5$.

E. Site-Level Voting

Table I indicates how ARTMAP voting, where voters decide on a prediction for each pixel, can boost performance by 3–4% on the prototype example. For mapping problems, however, a site or region fixes a more appropriate measurement scale than individual pixels. On the large-scale remote sensing tests in this section, voting at the site level, rather than the pixel level, proved to be the more successful method. For site-level voting, a number of fuzzy ARTMAP networks are trained on a given input set, each with the inputs presented in a different randomly chosen order. Each voter then predicts the vegetation class of each test set site, as in Section B. Finally, then, the class prediction for each site is taken to be the one made by the largest number of voters.

Site-level voting improves fuzzy ARTMAP performance on a variety of six variable (combo 1) tests, provided that low-confidence predictions are ruled inconclusive [Table III(C)]. Setting $\bar{\rho} = 0.87$ increases individual network performance from 44.7% to 46.4%, and voting further increases performance to 48.6%. When $\bar{\rho}$ is larger than 0.9 the confidence threshold is too high. For $\bar{\rho} = 0.92$, individual test set performance falls to 42%, then to 29% for $\bar{\rho} = 0.95$. Notably, voting performance remains at 50% even for $\bar{\rho} = 0.92$, where individual network performance drops to 42%. Evidently, when $\bar{\rho}$ becomes too large for an individual network, which would then label too many pixels as inconclusive, a number of voters can pool predictions to maintain accuracy. Thus, choosing an appropriate confidence threshold $\bar{\rho}$ and using a voting strategy appear to be mutually beneficial.

On 13 variable tests (combo 2) with $\bar{\rho} = 0,\ 5$, voters improve fuzzy ARTMAP performance to 60.0%, compared to an individual network average of 57.2%. Adding more voters usually had little effect on results. Since maximum likelihood computes order-independent parameters, this technique has no analogue of ARTMAP voting. Site-level voting thus widens the accuracy gap between maximum likelihood (56.5%) and fuzzy ARTMAP (60.0%).

F. Hybrid Fuzzy ARTMAP—Maximum Likelihood Classification System

The system variation with the best performance combines the predictions of trained fuzzy ARTMAP and maximum likelihood systems. The success of this method is due to the observation that the two classifiers tend to make predictive

TABLE IV
CONFUSION MATRICES: SIX VARIABLE TESTS

A. Fuzzy ARTMAP ($\bar{\rho}$=0.87, 5 voters) - 48.6% correct (diagonal)

Predicted vegetation class	**Actual vegetation classes**								
	mixed conifer	canyon live oak	coast live oak	chamise	scrub oak	red shanks	s. mixed chaparral	n. mixed chaparral	Total
mixed conifer	**11.0**	5.0							16.0
canyon live oak		**1.8**	0.2		0.1			0.3	2.4
coast live oak			**11.9**	0.1	2.7	1.7		6.8	23.2
chamise				**6.8**			1.7	2.2	10.7
scrub oak		0.2		0.6	**4.4**	2.7	0.1	4.3	12.3
red shanks			0.3		3.6	**2.2**	1.0	1.8	8.9
s. mixed chaparral							**2.8**	0.9	3.7
n. mixed chaparral			3.6	5.5	1.2	3.4	1.4	**7.7**	22.8
Total	11	7	16	13	12	10	7	24	100

B. Maximum likelihood (CT = -21.6) - 48.8% correct (diagonal)

Predicted vegetation class	**Actual vegetation classes**								
	mixed conifer	canyon live oak	coast live oak	chamise	scrub oak	red shanks	s. mixed chaparral	n. mixed chaparral	Total
mixed conifer	**11.0**	5.0							16.0
canyon live oak		**2.0**	2.0	1.0	1.0			1.0	7.0
coast live oak			**6.5**		1.0	1.0		2.0	10.5
chamise				**9.0**				4.4	13.4
scrub oak			3.0	1.0	**5.0**	2.0		7.0	18.0
red shanks			3.5		5.0	**6.0**		5.3	19.8
s. mixed chaparral				1.0			**7.0**	2.0	10.0
n. mixed chaparral			1.0	1.0		1.0		**2.3**	5.3
Total	11	7	16	13	12	10	7	24	100

C. Convex combination (γ = 0.6) - 50.6% correct (diagonal)

Predicted vegetation class	**Actual vegetation classes**								
	mixed conifer	canyon live oak	coast live oak	chamise	scrub oak	red shanks	s. mixed chaparral	n. mixed chaparral	Total
mixed conifer	**11.0**	5.0							16.0
canyon live oak		**2.0**	2.0		0.9			1.0	5.9
coast live oak			**7.5**		1.3	1.3		2.6	12.7
chamise				**7.0**			0.2	4.1	11.3
scrub oak			2.6	1.9	**6.1**	2.0		6.5	19.1
red shanks			2.4		3.7	**6.0**	0.1	3.5	15.7
s. mixed chaparral				1.0			**6.6**	1.9	9.5
n. mixed chaparral			1.5	3.1		0.7	0.1	**4.4**	9.8
Total	11	7	16	13	12	10	7	24	100

errors in somewhat different circumstances. Confusion matrices (Tables IV and V) compare fuzzy ARTMAP and maximum likelihood predictions with test set ground truth classifications. For example, in combo 1 tests (six variables), fuzzy ARTMAP [Table IV(A)] makes more errors trying to identify red shanks sites than does maximum likelihood [Table IV(B)]. Both classifiers do well on mixed conifer sites, but both do poorly on canyon live oak and northern mixed chaparral. An ideal hybrid system would choose the right decision when the two disagree, but designing such an optimal combination for a given problem would again require *a priori* knowledge of the test set. Of a variety of hybrid algorithms tested, all

TABLE V
CONFUSION MATRICES: 13 VARIABLE TESTS

A. Fuzzy ARTMAP ($\bar{\rho}$=0.0, 5 voters) - 60.0% correct (diagonal)

Predicted	**Actual vegetation classes**								
vegetation class	mixed conifer	canyon live oak	coast live oak	chamise	scrub oak	red shanks	s. mixed chaparral	n. mixed chaparral	Total
mixed conifer	**11.0**	3.0							14.0
canyon live oak		**3.0**			1.0				4.0
coast live oak			**9.1**		1.7		0.2	2.7	13.7
chamise				**5.4**			2.0	1.3	8.7
scrub oak		1.0	0.6		**4.9**	2.0	0.9	2.5	11.9
red shanks			2.0		1.9	**7.0**			10.9
s. mixed chaparral				0.5			**2.1**		2.6
n. mixed chaparral			4.3	7.1	2.5	1.0	1.8	**17.5**	34.2
Total	11	7	16	13	12	10	7	24	100

B. Maximum likelihood (CT = 10.0) - 56.5% correct (diagonal)

Predicted	**Actual vegetation classes**								
vegetation class	mixed conifer	canyon live oak	coast live oak	chamise	scrub oak	red shanks	s. mixed chaparral	n. mixed chaparral	Total
mixed conifer	**11.0**	3.0							16.0
canyon live oak		**4.0**			1.0			1.0	7.0
coast live oak			**11.0**		4.5			4.0	10.5
chamise				**9.0**				2.0	13.4
scrub oak			3.0		**3.5**	3.0		8.0	18.0
red shanks					3.0	**4.0**		1.0	19.8
s. mixed chaparral				1.0			**7.0**	1.0	10.0
n. mixed chaparral			2.0	3.0		3.0		**7.0**	5.3
Total	11	7	16	13	12	10	7	24	100

C. Convex combination (γ = 0.6) - 61.1% correct (diagonal)

Predicted	**Actual vegetation classes**								
vegetation class	mixed conifer	canyon live oak	coast live oak	chamise	scrub oak	red shanks	s. mixed chaparral	n. mixed chaparral	Total
mixed conifer	**11.0**	3.0							14.0
canyon live oak		**3.0**			1.0			1.0	5.0
coast live oak			**11.2**		3.4			3.0	17.6
chamise				**6.8**			0.1	1.1	8.0
scrub oak		1.0	0.3		**4.1**	2.0	0.7	5.2	13.3
red shanks			2.0		3.0	**7.0**			12.0
s. mixed chaparral				0.6			**4.4**	0.1	5.1
n. mixed chaparral			2.5	5.6	0.5	1.0	1.8	**13.6**	25.0
Total	11	7	16	13	12	10	7	24	100

showed some improvement over that of the individual systems. The hybrid that consistently gave best results took a convex combination of the two systems' site-level predictions, as follows.

To select from the eight vegetation classes, maximum likelihood generates a prediction for each of the pixels in a site. Those pixels for which a definitive prediction is made (i.e., not an "inconclusive" response) can form a vector with components equal to the fraction of definitive pixels in that site assigned to each of the eight classes. An analogous prediction vector for fuzzy ARTMAP lists the fraction of voters choosing each class. A convex combination of the two vectors, giving weight γ to fuzzy ARTMAP and $(1-\gamma)$ to maximum likelihood, forms the hybrid prediction vector.

To test performance improvement of a hybrid, fuzzy ARTMAP and maximum likelihood systems were chosen to maximize individual system performance accuracy. For combo 1, fuzzy ARTMAP with five voters and $\bar{\rho} = 0.87$ gave 48.6% correct predictions, while maximum likelihood with $\mathrm{CT} = -21.6$ was 48.8% correct [Table III(C)]. A convex-combination hybrid with $\gamma = 0.6$, which gives 60% weight to fuzzy ARTMAP, improved test set performance to 50.6%. With $\gamma = 0.4$, which gives 60% weight to maximum likelihood, performance was almost as high (50.4%). With $\gamma = 0.6$, the hybrid system allows maximum likelihood predictions of red shanks and southern mixed chaparral sites to dominate the distributed (and largely incorrect) fuzzy ARTMAP predictions for these classes [Table IV(C)]. For coast live oak sites and northern mixed chaparral, fuzzy ARTMAP compensates for a number of the maximum likelihood errors. At canyon live oak sites, where the two systems make the same errors, hybrid prediction is no better.

For combo 2, fuzzy ARTMAP with five voters and $\bar{\rho} = 0.0$ gave 60.0% correct predictions, while maximum likelihood with $\mathrm{CT} = 10.0$ was 56.5% correct (Table III-C). Since optimal performance of the two systems now differs by 3.5%, some hybrids do not give better predictions than fuzzy ARTMAP alone. Nevertheless, a convex combination with $\gamma = 0.6$ again gave the best performance, boosting accuracy to 61.1%. However, giving 60% weight to maximum likelihood $(\gamma = 0.4)$ brought performance back down to the level of maximum likelihood alone. On combo 2, with all 13 input variables, fuzzy ARTMAP performance on the difficult northern mixed chaparral class is greatly improved [Table V(A)]. Maximum likelihood shows less improvement [Table V(B)], and predictions of the convex combination fall between the two [Table V(C)].

IX. Conclusion

This paper provides an introduction to the fuzzy ARTMAP neural network in the context of remote sensing classification problems. Tests on a prototype remote sensing problem and an actual vegetation mapping problem illustrate a number of points. First, a voting strategy improves prediction by training several fuzzy ARTMAP networks on different orderings of an input set. This strategy assigns confidence estimates to competing predictions. Second, fuzzy ARTMAP and maximum likelihood perform differently for different combinations of input variables. Fuzzy ARTMAP performance increases using brightness, greenness, and wetness as compared to the original spectral bands, and increases even more when these are combined. Ancillary inputs improve maximum likelihood and fuzzy ARTMAP by similar amounts. Third, a hybrid fuzzy ARTMAP and maximum likelihood classification system can improve overall predictive accuracy since the two classifiers tend to make somewhat different predictive errors. Fourth, results from a group of pixels pooled together form accurate mappings across functional regions or sites, and site-level predictions are more useful than pixel-level predictions.

Acknowledgment

The authors would like to thank J. Franklin for providing the Cleveland Forest data set, S. Macomber for compiling the data, and J. Collins for help with the maximum likelihood computations.

References

[1] R. M. Hoffer and Staff, *Natural Resources Mapping in Mountainous Terrain by Computer Analysis of ERTS-1 Satellite Data*, Agricultural Experiment Station Research Bulletin 919 and LARS Contract Report 061575, Purdue University, W. Lafayette, IN, p. 124, 1975.

[2] A. H. Strahler, T. L. Logan, and N. A. Bryant, "Improving forest cover classification accuracy from Landsat by incorporating topographic information," in *Proc. 12th Int. Symp. Remote Sensing Environ.*, Ann Arbor, MI: Environmental Research Institute of Michigan, 1978, pp. 927–942.

[3] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy, "Conjugate-gradient neural networks in classification of multisource and very-high-dimensional remote sensing data," *Int. J. Remote Sensing*, vol. 14, pp. 2883–2903, 1993.

[4] A. Baraldi and F. Parmiggiani, "A neural network for unsupervised categorization of multivalued input patterns: An application of satellite image clustering," *IEEE Trans. Geosci. Remote Sensing*, vol. 33, pp. 305–316, 1995.

[5] H. Bischoff, W. Schneider, and A. J. Pinz, "Multispectral classification of Landsat images using neural networks," *IEEE Trans. Geosci. Remote Sensing*, vol. 30, pp. 482–490, 1992.

[6] P. D. Heermann and N. Khazenie, "Classification of multispectral remote sensing data using a back-propagation neural network," *IEEE Trans. Geosci. Remote Sensing*, vol. 30, pp. 81–88, 1992.

[7] Y. C. Tzeng, K. S. Chen, W.-L. Kao, and A. K. Fung, "A dynamic learning neural network for remote sensing applications," *IEEE Trans. Geosci. Remote Sensing*, vol. 32, pp. 1096–1103, 1994.

[8] S. E. Decatur, "Application of neural networks to terrain classification," in *Proc. Int. Joint Conf. Neural Networks*, Piscataway, NJ: IEEE, 1989, vol. I, pp. 283–288.

[9] S. Grossberg, E. Mingolla, and J. R. Williamson, "Synthetic aperture radar processing by a multiple scale neural system for boundary and surface representation," *Neural Networks*, vol. 8, pp. 1005–1028, 1995.

[10] S. Gopal, D. M. Sklarew, and E. Lambin, "Fuzzy-neural networks in multi-temporal classification of landcover change in the Sahel," in *Proc. DOSES Workshop New Tools Spatial Anal.*, Lisbon, Portugal, DOSES, EUROSTAT, pp. 55–68, 1994.

[11] A. Abuelgasim, S. Gopal, S. J. Irons, and A. Strahler, "Classification of ASAS multi-angle and multispectral measurements using artificial neural networks," *Remote Sensing Environ.*, vol. 59, pp. 79–87, 1996.

[12] G. F. Hepner, T. Logan, N. Ritter, and N. Bryant, "Artificial neural network classification using a minimal training set comparison of conventional supervised classification," *Photogram. Eng. Remote Sensing*, vol. 56, pp. 469–473, 1990.

[13] R. W. Fitzgerald and B. G. Lees, "Assessing the classification accuracy of multisource remote sensing data," *Remote Sensing Environ.*, vol. 47, pp. 362–368, 1994.

[14] R. G. Congalton, K. Green, and J. Teply, "Mapping old growth forests on national forest and park lands in the pacific northwest from remotely sensed data," *Photogram. Eng. Remote Sensing*, vol. 59, pp. 529–535, 1993.

[15] W. G. Cibula and M. O. Nyquist, "Use of topographic and climatological models in a geographic data base to improve Landsat MSS classification for Olympic National Park," *Photogrammetric Engineering and Remote Sensing*, vol. 53, pp. 67–75, 1987.

[16] T. D. Frank, "Mapping dominant vegetation communities in the Colorado Rocky Mountain front range with Landsat Thematic Mapper and digital terrain data," *Photogram. Eng. Remote Sensing*, vol. 54, pp. 1727–1734, 1988.

[17] J. Franklin, T. L. Logan, C. E. Woodcock, and A. H. Strahler, "Coniferous forest classification and inventory using Landsat and digital terrain data," *IEEE Trans. Geosci. Remote Sensing*, vol. 24, pp. 139–149, 1986.

[18] B. G. Lees and K. Ritman, "Decision-tree and rule-induction approach to integration of remotely sensed and GIS data in mapping vegetation in disturbed or hilly environments," *Environ. Mgt.*, vol. 15, pp. 823–831, 1991.

[19] A. K. Skidmore, "An expert system classifies Eucalypt forest types using Thematic Mapper data and a digital terrain model," *Photogram. Eng. Remote Sensing*, vol. 55, pp. 1449–1464, 1989.

[20] C. E. Woodcock, J. Collins, S. Gopal, V. Jakabhazy, X. Li, S. Macomber, S. Ryherd, Y. Wu, V. J. Harward, J. Levitan, and R. Warbington, "Mapping forest vegetation using Landsat TM imagery and a canopy reflectance model," *Remote Sensing Environ.*, vol. 50, pp. 240–254, 1994.

[21] D. E. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation," *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 318–362.

[22] P. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," Ph.D Thesis, Cambridge, MA: Harvard University, 1974.

[23] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, pp. 386–408, 1958. Reprinted in J. A. Anderson and E. Rosenfeld, Eds., *Neurocomputing: Foundations of Research*. Cambridge, MA: MIT Press, 1988, pp. 18–27.

[24] Y. Salu and J. Tilton, "Classification of multispectral image data by the binary diamond neural network and by nonparametric, pixel-by-pixel methods," *IEEE Trans. Geosci. Remote Sens.*, vol. 30, pp. 606–618, 1993.

[25] D. M. Foody, M. B. McCulloch, and W. B. Yates, "Classification of remotely sensed data by an artificial neural network: Issues related to training data characteristics," *Photogramm. Eng. Remote Sens.*, vol. 61, pp. 391–401, 1995.

[26] P. Gong, "Frequency-based contextual classification and grey-level vector reduction for land-use identification," *Photogramm. Eng. Remote Sens.*, vol. 58, pp. 439–448, 1992.

[27] A. K. Skidmore and B. J. Turner, "Forest mapping accuracies are improved using a supervised nonparametric classifier with SPOT data," *Photogramm. Eng. Remote Sens.*, vol. 54, pp. 1415–1421, 1988.

[28] P. C. Van Deusen, "Modified highest confidence first classification," *Photogramm. Eng. Remote Sens.*, vol. 61, pp. 419–425, 1995.

[29] S. Grossberg, "Adaptive pattern classification and universal recoding, II: Feedback, expectation, olfaction, and illusions," *Biolog. Cybernet.*, vol. 23, pp. 187–202, 1976.

[30] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vision Graphics Image Process.*, vol. 37, pp. 54–115, 1987.

[31] *Pattern Recognition by Self-Organizing Neural Networks*, G. A. Carpenter and S. Grossberg, Eds. Cambridge, MA: MIT Press, 1991.

[32] G. A. Carpenter and S. Grossberg, "ART 2: Stable self-organization of pattern recognition codes for analog input patterns," *Appl. Optic.*, vol. 26, pp. 4919–4930, 1987.

[33] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an Adaptive Resonance system," *Neural Networks*, vol. 4(6), pp. 759–771, 1991.

[34] G. A. Carpenter, S. Grossberg, and J. H. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, vol. 4, pp. 565–588, 1991.

[35] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. Neural Networks*, vol. 3, pp. 698–713, 1992.

[36] G. A. Carpenter and S. Grossberg, "ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures," *Neural Networks*, vol. 3, pp. 129–152, 1990.

[37] B. Moore, "ART 1 and pattern clustering," in *Proc. 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski, Eds. San Mateo, CA: Morgan Kaufmann, 1989, pp. 174–185.

[38] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[39] G. A. Carpenter and N. Markuzon, "ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases," *Neural Networks*, Tech. Rep. CAS/CNS TR-96-017, Boston, MA: Boston University, 1996.

[40] W. J. Matyas and I. Parker, "CALVEG mosaic of existing vegetation of California," San Francisco, CA: Regional Ecology Group, U.S. Forest Service, Region 5, 630 Sansome Street, p. 27, 1980.

[41] T. M. Lillesand and R. W. Kiefer, *Remote Sensing and Image Interpretation*, 3rd ed. New York: Wiley, 1994, pp. 594–596.

[42] J. Richards, *Remote Sensing Digital Image Analysis: An Introduction*. Berlin, Germany: Springer-Verlag: 1993, pp. 182–189.

[43] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[44] E. P. Christ, "A TM tasseled cap equivalent transformation for reflectance factor data," *Remote Sensing Environ.*, vol. 17, pp. 301–306, 1985.

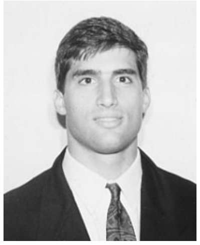
[45] S. E. Franklin, D. R. Peddle, and J. E. Moulton, "Spectral/geomorphometric discrimination and mapping of terrain: A study of Gros Morne National Park," *Canadian J. Remote Sensing*, vol. 15, pp. 28–42, 1989.

[46] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy, "Neural network approaches versus statistical methods in classification of multisource remote sensing data," *IEEE Trans. Geosci. Remote Sensing*, vol. 28, pp. 540–552, 1990.

Gail A. Carpenter received the B.A. degree from the University of Colorado, Boulder, and the M.A. and Ph.D. degrees from the University of Wisconsin, Madison, in mathematics.

She taught at the Mathematics Departments at Massachusetts Institute of Technology, Cambridge, and Northeastern University before moving to Boston University, where she is a Professor of Cognitive and Neural Systems and Professor of Mathematics. Her research includes the development, analysis, and application of neural models of vision, nerve impulse generation (Hodgkin-Huxley equations), synaptic transmission, biological rhythms, and distributed recognition codes; adaptive resonance theory (ART 1, ART 2, ART 2-A, ART 3, fuzzy ART, distributed ART) architectures for self-organizing category learning and pattern recognition; and systems that incorporate ART modules into neural network architectures for incremental supervised learning and applications (ARTMAP, fuzzy ARTMAP, ART-EMAP, ARTMAP-IC, distributed ARTMAP).

Dr. Carpenter serves on the editorial boards of *Brain Research,* IEEE TRANSACTIONS ON NEURAL NETWORKS, *Neural Computation*, and *Neural Networks.* She has been a member of the Governing Board of the International Neural Network Society since its founding, in 1987, and is also a member of the Council of the American Mathematical Society.



Marin N. Gjaja received the B.S.E. degree from Princeton University, Princeton, NJ, in 1991 and the Ph.D. degree from the Department of Cognitive and Neural Systems, Boston University, Boston, MA, in 1996.

He is currently employed with the Boston Consulting Group, Inc. His research interests include pattern recognition, classification, coding theory, and applications of neural networks for technology and business.

Sucharita Gopal received the Ph.D. degree from the Department of Geography, University of California, Santa Barbara, in 1989.

Since 1989 she has carried out research in the areas of spatial cognition, fuzzy sets, spatial accuracy, and geographical information systems. Over the last few years, she has conducted research in the applications of neural networks to landcover classification, change detection, and modeling in remote sensing. She is currently an Associate Professor of Geography and a member of the Center for Remote Sensing at Boston University.

Curtis E. Woodcock received the B.A., M.A., and Ph.D. degrees from the Department of Geography at the University of California, Santa Barbara.

Since 1984 he has taught at Boston University, where he is currently Associate Professor and Chair of Geography and a Researcher in the Center for Remote Sensing. His current research interests in remote sensing include mapping of forest structure and change, spatial modeling of images, inversion of canopy reflectance models, detection of environmental change, and issues of map accuracy.